



Tags y Releases

Transcripción

[00:00] Hola todos y todas. Bienvenidos y bienvenidas una vez más a este curso de Git. Ahora tenemos un proyecto finalizado, revisamos las modificaciones hechas a lo largo del proyecto, ya revisamos el código, entonces quiero generar una versión. Vamos por ejemplo a hacer `git log -n 2`.

[00:21] Con esto vamos a ver los últimos dos commits que hemos hecho. Y yo quiero, a partir del último commit, que a partir de ese último commit sea clavada una bandera, marcar un checkpoint como en los juegos de Sonic o Mario cuando pasabas por una bandera. Queremos marcar que en ese punto tenemos la versión por ejemplo 0.1.

[00:42] En ese commit marcamos el momento que lanzamos esa versión 0.1 o marcamos cualquier momento que queramos, en realidad. Quiero definir ese commit con un punto cualquiera y que yo le estoy dando como ejemplo una versión. ¿Cómo puedo hacer eso? En otros sistemas de control, de contenido, no de versión, existe un concepto de tags.

[01:06] Por ejemplo si vas a crear un blog, uno tiene tags para marcar posts específicos con categorías. En Git podemos usar un concepto parecido también llamado tag. Un tag marca un punto en nuestra aplicación, un punto que no puede ser modificado, fijo, por ejemplo el lanzamiento de una versión y la versión 0.1, una vez lanzada nunca más es modificada. Cualquier modificación que yo quiera agregar va a ser agregada a la 0.2 y no a la 0.1.

[01:42] Entonces la tag sirve exactamente para eso, para definir un punto que no cambia nunca más en nuestra aplicación. ¿Pero entonces, Bruno, eso significa que voy a hacer que nuestro código nunca más sea editable? No, no es eso. Solo voy a crear un punto al que puedo llegar y tener un código exactamente en ese estado.

[02:02] Voy a estar seguro de que nuestro código va a estar en ese estado de la versión 0.1. Bueno, vamos a crear entonces esa versión 0.1. Voy a hacer un clear, enter, y ya sabemos que estamos hablando de tag entonces, vamos a comenzar colocando git tag. ¿Cómo creo esa tag? Necesito informar a Git que estoy agregando una tag.

[02:26] Entonces lo que voy a hacer es utilizar, perdón tag con G ahí, entonces como le tengo que decir a Git que estoy agregando una tag, vamos a utilizar el parámetro -a, de add en inglés. Y ahora necesito dar un nombre a esa tag. Ese nombre puede ser el que sea. Puedo ponerle cocodrilo si quiero, puedo ponerle versión -0.1 por ejemplo.

[02:56] Pero en este caso yo voy a utilizar la nomenclatura v0.1.0. Esa va ser nuestra versión. ¿Bruno, eso es una regla? ¿Tengo que poner esa tag así siempre? No, solo sugiero ponerlo de esa forma, pero si ustedes se sienten más cómodos usando otra nomenclatura, no hay problema. Pueden usarla.

[03:17] También puedo agregar un mensaje para ese tag con el parámetro -m, como hacemos en los commits. Entonces por ejemplo usamos git tag -a v0.1.0 -m "Lanzando la primera versión". Vamos a colocar entre paréntesis (BETA) de la aplicación de cursos. Okay, damos un enter. Parece como si no hubiera pasado nada, pero en realidad ya creé una tag, un punto en nuestra aplicación.

[03:59] Si yo ejecuto ahora git tag, enter, vemos que nos aparecen todas las tags que en este caso es solamente la v0.1.0. Ahora, una cosa interesante que es donde yo quería llegar. Saben que yo puedo hacer push de master o de

cualquier branch a donde yo quisiera. Por ejemplo, puedo hacer un git push servidorlocal master, enter.

[04:39] Perfecto. Hicimos un push a nuestro servidor local. Y puedo también enviar mi tag, puedo hacer un git push servidorlocal v0.1.0. Damos enter, y vean que se creó una nueva tag. Entonces esto significa que puedo enviar mi tag al servidor local.

[05:01] Veán que ahora en el servidor tenemos esa nueva versión, esa versión de 0.1.0. Y ahora, el punto más interesante. Voy a hacer un clear a la pantalla y vamos a hacer un git remote -v, enter. Veán que nosotros, además de nuestro servidor local, también tenemos nuestro origin que es nuestro servidor de GitHub.

[05:28] Entonces vamos a enviar nuestra master a GitHub, vamos a hacer un git push origin master, enter. Voy a parar acá el comando porque parece que no está funcionando. Un segundo. Vamos a intentar de nuevo, git push origin master. Perfecto, ahora sí. Se envió nuestro master, y ahora voy a enviar nuestra tag así como hicimos con el servidor local. Entonces git push origin v.0.1.0. Perfecto.

[06:08] Vamos a ver. Bien, ahí se está enviando y genial. Enviamos nuestra nueva tag para GitHub. Y ahora, ¿cómo visualizo tags en GitHub? Bueno, ese es un punto interesante donde quería llegar. Vamos a ir a GitHub, a nuestra página de GitHub, vamos a nuestro curso de Alura y vean que ahora ya tengo 17 commits hechos y vean una cosa, acá tengo un tag.

[06:39] Tengo entonces acá todos los commits que hicimos, modificando el nombre del curso, fue el último commit, aparecen las branches y acá tenemos un tag. Fíjense que también lo pueden ver a través de acá, de releases. Si nosotros cliqueamos ahí, vemos que tenemos ese tag v0.1.0, si hacemos clic en los puntitos nos dice "lanzando la primera versión (BETA) de la aplicación".

[07:06] Nos está diciendo en cuál branch fue hecho, nos está diciendo también cuál fue el último commit, el commit hecho en esa tag. Volvemos para atrás y vean también por ejemplo que tenemos un .zip, entonces nosotros podemos bajar nuestra aplicación como un zip y compartirla con otras personas.

[07:30] Entonces por ejemplo vamos a hacer eso, vamos a bajarla. Tenemos acá a Alura, y acá tenemos nuestro Alura 1.0, por ejemplo vamos a hacerle un clic ahí, doble clic a nuestro index.html. Un segundo, antes de hacer eso, vamos a extraerlo en downloads, acá. Perfecto. Y vamos a descarga, acá tenemos nuestro zip, acá tenemos nuestro archivo de Alura, y si hacemos un clic derecho, abrir con, vamos a elegir otra aplicación, acá vamos a utilizar Visual Studio Code.

[08:16] Bien. Aceptar. Y vean, acá está el archivo nuevo, el que bajé en el .zip. Miren, está igual al archivo que nosotros teníamos antes en Bruno. Entonces con esto generamos una versión para nuestra aplicación, un punto histórico que puede ser enviado incluso a GitHub u otra persona que quiera usar nuestra aplicación.

[08:39] Y esa persona por ejemplo puede siempre ver cuáles son nuestras últimas versiones generadas con las tag utilizando esa URL, que cada uno de nosotros vamos a tener una URL diferente. Esto es muy bueno, es una feature muy importante, es una característica muy importante de GitHub para tener en cuenta.